TECHNICAL RESEARCH REPORT

Channel--Matched Hierarchical Table--Lookup Vector Quantization for Transmission of Video Over Wireless Channels

by H. Jafarkhani, N. Farvardin

T.R. 97-19

INSTITUTE FOR SYSTEMS RESEARCH

Sponsored by
the National Science Foundation
Engineering Research Center Program,
the University of Maryland,
Harvard University,
and Industry

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **1997**	2. REPORT TYPE	3. DATES COVERED **00-00-1997 to 00-00-1997**

4. TITLE AND SUBTITLE **Channel-Matched Hierarchical Table-Lookup Vector Quantization for Transmission of Video Over Wireless Channels**	5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Department of Electrical Engineering,Institute for Systems Research,University of Maryland,College Park,MD,20742**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S) 11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT
see report

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**6**	

Channel–Matched Hierarchical Table–Lookup Vector Quantization for Transmission of Video Over Wireless Channels *

Hamid Jafarkhani
Nariman Farvardin
Department of Electrical Engineering and
Institute for Systems Research
University of Maryland
College Park, MD 20742

ABSTRACT

We propose a channel-matched hierarchical table-lookup vector quantizer (CM-HTVQ) which provides some robustness against the channel noise. We use a finite-state channel to model slow fading channels and propose an adaptive coding scheme to transmit a source over wireless channels. The performance of CM-HTVQ is in general slightly inferior to that of channel-optimized vector quantizer (COVQ) (the performances coincide at some cases); however, the encoder complexity of CM-HTVQ is much less than the encoder complexity of COVQ.

1.INTRODUCTION

Vector quantization is a powerful tool for source coding which has been used in many speech and image coding systems [1]. The encoder of a vector quantizer (VQ) is usually implemented by computing the distortion between the input vector and each codevector in the codebook and finding the codevector which results in minimum distortion. The decoder however is a simple table lookup. In [2], Chang et al. have proposed a hierarchical table-lookup vector quantizer (HTVQ) in which the encoder is implemented using a table-lookup in multiple stages. For every possible input vector, the table-lookups store the codeword of the nearest codevector. Then, in the process of encoding, input vectors to the encoder are used directly as addresses for the tables to select the best codeword. The entire idea is predicated on the assumption that there are a finite number of input vectors, i.e., each input sample is already quantized – a reasonable assumption in most practical situations. Assuming n-bit input samples and a k-dimensional VQ, the size of such a table is 2^{nk} times the number of output bits. To make sure that the table sizes do not exceed manageable limits for large dimension VQ's, [2] adopts a hierarchical structure as shown in Fig. 1. The input address of Table $i+1$ in Fig. 1 is constructed by combining k_{i+1} outputs of Table i. The most practical configuration is $k_i = 2\ \forall i$ which is assumed throughout this paper. When applied to speech coding, the system in [2] is shown to suffer only a 1 dB performance degradation compared with a full-search VQ while the encoding complexity is dramatically reduced. Vishwanath and Chou have used the same system for video coding [3]. They also combine HTVQ with a discrete wavelet transform to improve the performance of the HTVQ. They demonstrate the desirable characteristics of their system for "interactive multicast over multiple rate channels."

In this paper, we study the performance of the scheme proposed by Vishwanath and Chou over noisy channels and propose a channel–matched HTVQ (CM-HTVQ) which provides the simplicity of an HTVQ, i.e., table-lookup encoding and decoding, and robustness against channel noise. We present results for memoryless Gaussian sources, still images, and video over memoryless binary symmetric channels (BSC's) and wireless channels with slow lognormal fading. Section 2 provides the design procedure of a CM-HTVQ for BSC's and presents numerical results. In Section 3, the results of Section 2 are extended to a finite-state channel (FSC). Section 4 provides concluding remarks.

2.CM-HTVQ FOR BSC'S

The basic idea behind channel-optimized VQ (COVQ) is to design the VQ encoder and decoder such that the end-to-end average distortion after encoding, transmission over the channel, and decoding is minimized [4].

*Prepared in part through collaborative participation in the Advanced Telecommunications/Information Distribution Research Program (ATIRP) Consortium sponsored by the U.S. Army Research Laboratory under Cooperative Agreement DAAL01-96-2-0002 and in part through support received from the National Institute of Standards and Technology grant 70NANBSH0051.

The optimization is done for a given source, a given noisy channel, a fixed dimension k, and a fixed codebook size M. The design algorithm is a modified generalized Lloyd algorithm, with the following expressions for the optimal partition, consisting of encoding cells, S_i's, and optimal codevectors, $\mathbf{c}_i$'s (squared-error distortion measure is assumed) [4]:

$$S_i = \{\mathbf{x} : \sum_{j=0}^{M-1} P(j|i) \parallel \mathbf{x} - \mathbf{c}_j \parallel^2 \leq \sum_{j=0}^{M-1} P(j|l) \parallel \mathbf{x} - \mathbf{c}_j \parallel^2,\ \forall l\}, \tag{1}$$

and

$$\mathbf{c}_j = \frac{\sum_{i=0}^{M-1} P(j|i) \int_{S_i} \mathbf{x} p(\mathbf{x}) d\mathbf{x}}{\sum_{i=0}^{M-1} P(j|i) \int_{S_i} p(\mathbf{x}) d\mathbf{x}}, \tag{2}$$

where $P(j|i)$ denotes the probability that j is received given that i is transmitted, $p(\mathbf{x})$ is the k-fold probability density function of the source, and $i, j \in \{0, 1, \cdots, M-1\}$. It is a well known fact that the performance of COVQ is better than that of VQ when the channel is noisy.

Like a regular VQ, a COVQ suffers from the high encoding complexity problem. To use the hierarchical structure of an HTVQ and design a COVQ which uses lookup tables, first we consider the case of an N-stage HTVQ and a memoryless BSC with a crossover probability of ϵ (the transition probability $P(j|i)$ can be computed when ϵ is known).

Since the purpose of the first $N-1$ stages is to provide the best pair of addresses for the N^{th} stage, the design procedure for these stages remains unchanged. The last stage however must be adapted to the characteristics of the channel. So, we use a COVQ to design the lookup table of the last stage. We call this structure CM-HTVQ because the last stage of the HTVQ is matched to the characteristics of the channel. Note that CM-HTVQ is not optimal since the design of the intermediate stages, even for noiseless channels, is not optimal [2].

Tables 1 and 2 show the performance of HTVQ, VQ, CM-HTVQ, and COVQ for a memoryless Gaussian source at rates 0.5 and 1 bit/sample (compression ratio (CR) 16:1 and 8:1), respectively. We have used simulated annealing to assign indices [5] to the codevectors of VQ and HTVQ. The source consists of samples from a memoryless Gaussian source quantized using an 8-bit Lloyd-Max scalar quantizer. As it is clear from the tables, the difference between the performance of COVQ and CM-HTVQ is less than the gap between VQ and HTVQ for a noiseless channel ($\epsilon = 0.0$). The gap shrinks as the channel becomes noisier. The same observation can be made based on the simulation results for the 512×512 Lenna (Tables 3 and 4). In this case, our quantizers are designed using five USC images (Couple, Crowd, Man, Woman1, and Woman2) as the training sequence. The simulation results are obtained by computing the average results of 10 experiments.

Similar results are obtained for the 360×288 "Salesman" video sequence, Table 5. To quantize a video sequence, we use two quantizers. One of them quantizes frame differences and the other one compresses the original frames every 5 frames. To quantize the original frames, we use the quantizer designed for still images at a bit rate $r = 1$ bit per pixel (bpp). The other quantizer is designed using ten frame differences obtained from the "Miss America" video sequence and operates at a bit rate $r = 0.5$ bpp.

It is also clear from the tables that CM-HTVQ demonstrates some robustness against the channel noise (compared with HTVQ and VQ). Thus, for the case of a memoryless BSC, not only does CM-HTVQ provide a simple table-lookup encoding, but also it achieves almost the same performance as that of COVQ.

3.CM-HTVQ FOR FADING CHANNELS

To tackle the same problem for a wireless channel, we model the channel by an FSC with a BSC associated with each state and a Markov chain governing the transition between the states [6]. A simple way to model a slowly-varying fading channel by an FSC is to divide the range of the received signal-to-noise ratios (SNR's) into a number of equiprobable intervals, representing the states, and model each state by a BSC. The details for a lognormal fading channel employing uncoded binary phase-shift keying as the signaling system can be found in [7]. In this work, we assume that both the transmitter and receiver have access to the channel state information (CSI). Repeating the above design procedure for each of the constituent channels of the FSC and using the appropriate table based on the available CSI lead to a CM-HTVQ system which is matched to the fading channel (Fig. 2).

The main advantage of HTVQ, other than table-lookup encoding, is the simplicity of transcoding from a higher bit rate to a lower bit rate code. This can be done by simply adding new stages to the HTVQ, basically amounting to adding memory to the hardware. One important application in which this table-lookup transcoding might prove useful is multicasting. Let us think of the connection between the consecutive stages of HTVQ as a channel. These channels might be noiseless channels, memoryless BSC's, or slowly-varying wireless channels. Such a model supports a broad range

of scenarios such as the example provided in Fig. 3, illustrating a "heterogeneous" network in which the transmitter sends a bit stream which is compressed using an HTVQ with a CR of 4:1. At Node A, the bit stream is retransmitted to Nodes B and C. The channel between Nodes A and B is a lognormal fading channel, modeled by an FSC, operating at a bit rate corresponding to a CR of 8:1; the channel between Nodes A and C is a memoryless BSC with bit error rate $\epsilon_1 = 0.1$ operating at a bit rate corresponding to a CR of 4:1. The goal is to design a system which is capable of transcoding the bit stream by table-lookups (without an expensive decoding followed by encoding) at Node A which provides robustness over the noisy channels of links AB and AC. Note that, in general, the system must be able to handle any other rate reduction or channel combination only by using lookup tables.

Our approach is to use COVQ's to design the CM-HTVQ encoding and decoding tables; the encoding tables are then used at the origination nodes for transcoding and the decoding tables are used at the destination nodes for the decoding. To explain this concept, let us concentrate on the example of Fig. 3 where we need a 2-stage HTVQ to transmit the source to Node A. Since the output of the first stage is directly used as the address for the second stage, the imaginary channel between the two stages is noiseless. Also, the channel connected to Node A is a noiseless channel; so, a regular HTVQ with a CR of 4:1 can be used to transmit the source to Node A. At Node A, we need to generate two bit streams, one for transmission to Node B and another one for Node C. The transcoding is performed using lookup tables designed as follows.

Let us assume that the average received SNR of the fading channel in link AB is 10 dB and we use an FSC with two states to model the channel (Channel 1 in [7]). As it is described in [7], the FSC model consists of a BSC with probability of error equal to 0.1 and a noiseless channel. State probabilities are 0.238 and 0.762, respectively. Two 8-dimensional COVQ's with a CR of 8:1 are designed for the two constituent BSC's of the FSC and a lookup table is designed for each BSC. To transmit the sequence over link AB, at each time instant, the appropriate encoding table is used based on the CSI (Fig. 2). Likewise, a 4-dimensional COVQ with a CR of 4:1 is designed for a memoryless BSC with probability of error equal to $\epsilon_1 = 0.1$. Such a COVQ is used to design the encoding table for transcoding the sequence for link AC. The table provides a new encoding map that is matched to the BSC. Note that this table is not used to compress the bit stream (because of the specific rate assumptions in the example), rather it is used to match the encoded sequence to the characteristics of the communication link. So, the number of output bits for this table is equal to the number of input bits. Such a table is not restricted to HTVQ and can be used for regular VQ's as well.

The decoding table associated with the appropriately designed COVQ must be used for decoding at Nodes B and C. Table 6 provides the received peak SNR (PSNR) at each node for the 512×512 Lenna. Also, we report the results for the case that the tables are not matched to the characteristics of the channel. Note that the result at Node C is different from the performance of a system which consists of a 2-stage CM-HTVQ (designed for the BSC in link AC) and a hypothetical BSC connecting the source to Node C directly. This is because in the the example of Fig. 3, we need two reconstructed versions of the source at Nodes A and C. Our simulation results indicate that we only pay 0.14 dB penalty in the performance at Node C to generate a replica of the source at Node A. Also, note that if we do not use the "matching table" at Node A and transmit the received bit stream at Node A directly to Node C, the PSNR result at Node C degrades by about 9.1 dB. If we use regular VQ's for quantization, we can still utilize a 256-byte matching table at Node A and improve the performance by about 9.19 dB (Table 6). The result would be the same if we decoded the sequence at Node A and re-encoded it using a COVQ. So, the matching table provides the optimal bit stream by using simple lookup tables instead of decoding and encoding the sequence. Again our simulations indicate that if we wish to use a regular VQ, creating two versions of the source at Nodes A and C degrades the result by only about 0.05 dB compared to the case that we only generate the output at Node C. Similar results have been obtained for other channels we have studied.

4.CONCLUSIONS

We have proposed a CM-HTVQ which outperforms HTVQ when the channel is noisy. Also, we have used an FSC to model slow fading channels and proposed an adaptive coding scheme by using a set of encoder and decoder tables. The performance of CM-HTVQ is slightly inferior to that of COVQ (its complexity is much less); however, the performance gap between CM-HTVQ and COVQ is much less than the performance gap between HTVQ and VQ. The gap shrinks as the channel becomes noisier. CM-HTVQ can also be designed for struc-

tured VQ's like tree-structured VQ, multi-stage VQ, and finite-state VQ. Using table-lookups for the encoder of different types of VQ's, assuming a noiseless channel, has been studied in [8]. To design a channel-matched version, the VQ which is used for designing the last stage must be substituted with a COVQ. Future work includes the transmission over a wireless channel when the CSI is only available at the decoder. For this case, we have found an optimal solution for COVQ's. The same approach has been applied to the CM-HTVQ's. [1]

References

[1] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression.* Kluwer Academic Publishers, Boston, 1992.

[2] P. C. Chang, J. May, and R. M. Gray, "Hierarchical Vector Quantizers with Table-Lookup Encoders," *Int. Conf. Acoustics Speech and Signal Proc.,* pp. 1452–1455, 1985.

[3] M. Vishwanath and P. Chou, "An Efficient Algorithm for Hierarchical Compression of Video," *Int. Conf. Image Proc.,* vol. 3, pp. 275–279, Nov. 1994.

[4] N. Farvardin and V. Vaishampayan, "On the Performance and Complexity of Channel-Optimized Vector Quantizers," *IEEE Trans. Inform. Theory*, vol. 37, no. 1, pp. 155–160, Jan. 1991.

[5] N. Farvardin, "A Study of Vector Quantization for Noisy Channels," *IEEE Trans. on Inf. Theory*, vol. 36, pp. 799–809, July 1990.

[6] H. S. Wang and N. Moayeri, "Finite-State Markov Channel – A Useful Model for Radio Communication Channels," *IEEE Trans. on Vehicular Tech.*, vol. 44, no. 1, pp. 163–171, February 1995.

[7] H. Jafarkhani, P. Ligdas, and N. Farvardin, "Adaptive Rate Allocation in a Joint Source/Channel Coding Framework for Wireless Channels," *Vehicular Technology Conf.,* Atlanta, GA, pp. 492–496, Apr. 1996.

[8] N. Chaddha, P. Chou, and R. M. Gray, "Constrained and Recursive Hierarchical Table–Lookup Vector Quantization," *Data Compression Conf.*, Snowbird, UT, pp. 220–229, Apr. 1996.

[1]The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Goverment.

ϵ	VQ	COVQ	HTVQ	CM-HTVQ
0.00	2.33	2.33	1.18	1.18
0.01	1.91	2.02	0.93	1.04
0.03	1.25	1.67	0.49	0.88
0.05	0.74	1.43	0.15	0.77
0.10	-0.12	1.00	-0.45	0.55

Table 1: Memoryless Gaussian Source (SNR in dB); CR=16:1; block size=16.

ϵ	VQ	COVQ	HTVQ	CM-HTVQ
0.00	4.85	4.85	4.05	4.05
0.01	3.82	4.21	3.22	3.57
0.03	2.42	3.59	2.02	3.08
0.05	1.45	3.14	1.19	2.72
0.10	0.00	2.26	-0.14	1.98

Table 2: Memoryless Gaussian Source (SNR in dB); CR=8:1; block size=8.

ϵ	VQ	COVQ	HTVQ	CM-HTVQ
0.00	30.49	30.49	29.69	29.69
0.01	24.05	28.24	23.79	27.79
0.03	20.02	26.40	19.94	26.35
0.05	17.99	25.22	18.01	25.13
0.10	15.27	23.02	15.26	22.99

Table 3: Lenna (PSNR in dB); CR=16:1; block size=4×4.

ϵ	VQ	COVQ	HTVQ	CM-HTVQ
0.00	32.47	32.47	31.79	31.79
0.01	24.13	29.48	23.97	29.20
0.03	19.90	27.46	19.88	27.32
0.05	17.89	26.09	17.83	26.00
0.10	15.12	23.40	15.06	23.35

Table 4: Lenna (PSNR in dB); CR=8:1; block size=4×2.

ϵ	VQ	COVQ	HTVQ	CM-HTVQ
0.00	31.09	31.09	29.48	29.48
0.01	25.92	29.42	25.15	28.79
0.03	21.72	28.32	21.49	27.72
0.05	19.49	27.06	19.36	26.58
0.10	16.09	24.18	16.08	23.98

Table 5: Salesman sequence (average PSNR in dB for 100 frames); $r = 0.6$ bpp.

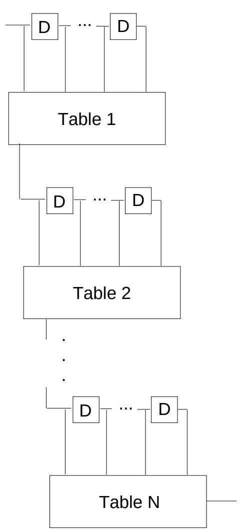

Figure 1: HTVQ Encoder.

Node	VQ	COVQ	HTVQ	CM-HTVQ
A	35.80	35.80	35.18	35.18
B	21.11	28.18	21.01	27.95
C	15.04	24.23	15.04	24.14

Table 6: Lenna (PSNR in dB).

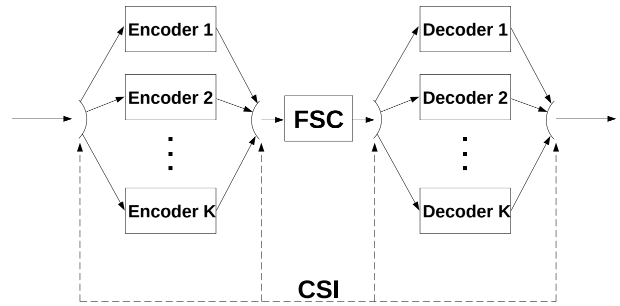

Figure 2: CM-HTVQ for FSC's.

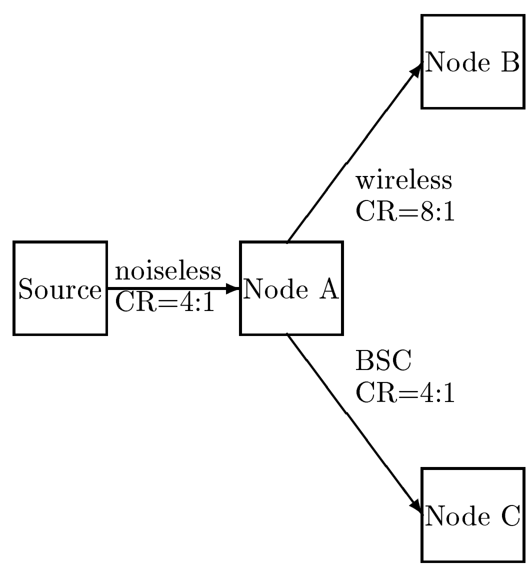

Figure 3: Example.